



Intrinsically Defined Planar Curves based on Explicit B-spline Curvature Functions

Norimasa Yoshida¹  and Takafumi Saito² 

<https://www.overleaf.com/project/6008155821daba0ce96a066d> ¹Nihon University, norimasa@acm.org
²Tokyo University of Agriculture and Technology, txsaito@cc.tuat.ac.jp

Corresponding author: Norimasa Yoshida, norimasa@acm.org

Abstract. In the design of aesthetic objects, controlling the curvature variation of a curve segment is an important task. Freeform curves, such as Bézier curves or B-spline curves, are widely used in many CAD systems. Controlling the curvature variation of freeform curves, however, is not easy because how the curvature behaves by moving its control points is not predictable. This paper introduces an intrinsically defined planar curve based on an explicit polynomial B-spline curve and its G^1 and G^2 Hermite interpolation method. The advantage of using explicit B-spline curvature functions instead of explicit Bézier curvature functions is that a wider variety of curvature variation can be represented by increasing the number of segments and modifying the knots. The proposed method can theoretically match any G^2 Hermite conditions. In the proposed approach, the curvature plot is specified in terms of an explicit B-spline curve and the explicit B-spline curve is integrated to generate a curve segment such that given G^1 or G^2 Hermite interpolation conditions are satisfied. As an application of the proposed curve, we show a method of controlling the curvature variation interactively by modifying *control curvatures* shown on the curvature comb satisfying G^1 or G^2 Hermite conditions.

Keywords: intrinsically defined curves, explicit B-spline curvature functions, G^1 and G^2 Hermite interpolation, curvature comb

DOI: <https://doi.org/10.14733/cadaps.2022.152-163>

1 INTRODUCTION

Freeform curves, such as Bézier curves or B-spline curves, are commonly used not only in CAD software but also in many other applications including graphics design software. Although freeform curves are widely used, it is not easy to control the curvature variation of freeform curves. Controlling the curvature of freeform curves by moving the control points is not easy due to the complexity of curvature computation. In this paper, we propose intrinsically defined curves based on explicit polynomial B-spline curvature functions. We

use an explicit B-spline curve [16] because it is a natural extension of a linear function. Although numerical integration and an optimization (of two parameters) is necessary to perform G^1 or G^2 Hermite interpolation, a curve segment can be generated in real time. As an application of the proposed curve, we present a method of controlling the curvature variation interactively by modifying *control curvatures* shown on the curvature comb satisfying G^1 or G^2 Hermite conditions.

The advantages of proposed curves over previous work are the following:

- Given G^1 or G^2 Hermite conditions, various kinds of curves with the control of curvature variation can be generated.
- Since B-spline curvature functions include linear and constant curvature functions, the proposed curve can theoretically match any G^2 Hermite conditions satisfying the monotonicity of the curvature and the curvature variation is still under control.
- To control the curvature variation, control curvatures on the curvature comb can be directly controlled satisfying G^1 or G^2 Hermite conditions.

The rest of this paper is organized as follows. In section 2, we review the related work on controlling the curvature variation both for freeform curves and for intrinsically defined curves. In Section 3, we describe the method for computing intrinsically defined curves based on explicit B-spline curvature functions. We show that an integration of an explicit B-spline curve is straightforward once it is converted to explicit Bézier curves. In Section 4, we present a G^1 or G^2 Hermite interpolation method using an optimization of two parameters. In Section 5, we show the results of G^1 and G^2 Hermite interpolation in addition to examples of controlling the curvature variation. In Section 6, we conclude our paper.

2 RELATED WORK

In computer aided design, curves are usually designed by freeform curves, such as Bézier curves or B-spline curves. Various work has been done for controlling the curvature variation of freeform curves. In quadratic polynomial Bézier curves, Sapidis et al. clarified specific position of control points where the curvature is monotonically varying [15]. Because of its simplicity that it has only one curvature extrema, quadratic Bézier curves are used for style design [18] and illustration [22].

Since quadratic Bézier curves cannot represent an inflection point and its curvature function is rather fixed, freeform curves of degree 3 or higher are widely used in many applications, such as illustration software and CAD/CAM systems. For curves of degree 3 or higher, the position of control points where the curvature is monotonically varying is not known except for typical class A Bézier curves [4]. Yoshida et al. showed that typical class A Bézier curves get closer to logarithmic spirals as the degree gets higher [23]. Wang proposed a shape control method of Bézier and B-spline curves by finding a solution of an optimization subject to monotone curvature variation constraints [20]. Li et al. have proposed a target curvature driven fairing algorithm for planar cubic B-spline curves [8]. Dietz et al. have numerically considered a cubic Bézier curve with monotonically varying curvature that satisfies G^2 Hermite conditions in [2] and a rational cubic Bézier curve in [3]. Habib et al. derived admissible regions for the G^2 Hermite spiral segment of a rational cubic Bézier curve [6]. Due to the complexity of computing curvature of freeform curves, controlling the curvature variation of freeform curves is not easy in general.

Most of the work related to intrinsic equations have dealt with piecewise linear curvature [12, 13, 19] with several exceptions: Watanabe et al. proposed intrinsically defined curves based on cubic Bézier curves [21]. The generalised Cornu spiral [1] has rational linear curvature profile, thus can be considered as the generalization of Cornu (Clothoid) and logarithmic spirals. Log-aesthetic curves [7, 10, 24], which has a shape parameter α , are the generalization of Clothoid curve ($\alpha = -1$), Nielsen's spiral ($\alpha = 0$), logarithmic spiral ($\alpha = 1$), the circle involute ($\alpha = 2$) and a circle ($\alpha = \pm\infty$). Log-aesthetic curves are high quality curves

whose logarithmic curvature graphs are linear. In all of the previous work, the shapes of curvature functions are rather limited. We provide a general framework that can represent various curvature functions by using an explicit B-spline curves.

Concerning G^2 Hermite interpolation, Meek et al. used Clothoids and circular arcs and showed that the method can match any G^2 Hermite conditions [9]. The proposed curves in this research have more variety of curvature functions than [9]. Miura et al. have used log-aesthetic curves for G^2 Hermite interpolation [11]. The quality of the generated curves may be high, but it is not clear that the method can match a wide variety of G^2 Hermite conditions.

In this work, we extend our previous work [25] by using explicit B-spline curvature functions instead of explicit Bézier curvature functions. The advantage of using explicit B-spline curvature functions is that a wider variety of curvature variation can be represented. For example, as shown in Fig. 8 (d), the proposed method can represent a curvature function like a step function. This kind of curvature function is necessary to cover the viable G^2 Hermite conditions. Since explicit B-spline curvature functions can represent both linear and constant curvature functions, the proposed method can theoretically match any G^2 Hermite data [9].

3 CURVES BASED ON EXPLICIT B-SPLINE CURVATURE FUNCTIONS

This section describes a method for computing an intrinsically defined curves based on an explicit polynomial B-spline curvature function. To simplify the description, we consider a curve in the standard form where the starting point and its tangent are the origin and $[0 \ 1]^T$, respectively. A curve in a general position can be obtained by performing an appropriate transformation to the curve in the standard form.

Let s be the arc length and $\kappa(s)$ be the curvature function. The tangential angle $\theta(s)$ is

$$\theta(s) = \int_0^s \kappa(u) du. \quad (1)$$

The curve position $\mathbf{P}(s)$ is

$$\mathbf{P}(s) = \begin{bmatrix} \int_0^s \cos(\theta(u)) du \\ \int_0^s \sin(\theta(u)) du \end{bmatrix}. \quad (2)$$

Eqs. (1), (2) are standard equations in differential geometry. If we use the curvature function of log-aesthetic curves [24] as $\kappa(s)$, Eq. (2) immediately becomes the equation of log-aesthetic curves. In this work, we specifically consider the case where $\kappa(s)$ is an explicit B-spline curve.

An explicit polynomial B-spline curve of degree n with control points $p_i (i = 1, 2, \dots, m)$ and knots $t_1, t_2, \dots, t_{m+n-1}$ in polar form [14, 16] is a curve whose x coordinate is the parameter t and y coordinate is

$$p(t) = \sum_{i=1}^m B_i^n(t) p_i \quad (t \in [t_n, t_m]) \quad (3)$$

Here, $B_i^n(t)$ is a B-spline basis function. In explicit B-spline curves, t coordinate c_i for each control point p_i , known as Greville abscissa, is the average of n consecutive knots [16]:

$$c_i = \frac{t_i + t_{i+1} + \dots + t_{i+n-1}}{n}. \quad (4)$$

Fig. 1 (a) shows an explicit cubic B-spline curve with knots $[0, 0, 0, \frac{1}{2}, 1, 1, 1]$ and control points $p_i (i = 1, \dots, 5)$. For an explicit polynomial B-spline curve, integration of Eq. (1) is straightforward once it is converted to explicit Bézier curve segments by appropriately inserting knots. Fig. 1(b) shows the Bézier control points $q_i (i = 1, 2, \dots, 7)$ and the control polygon converted from (a). Since the integration of an

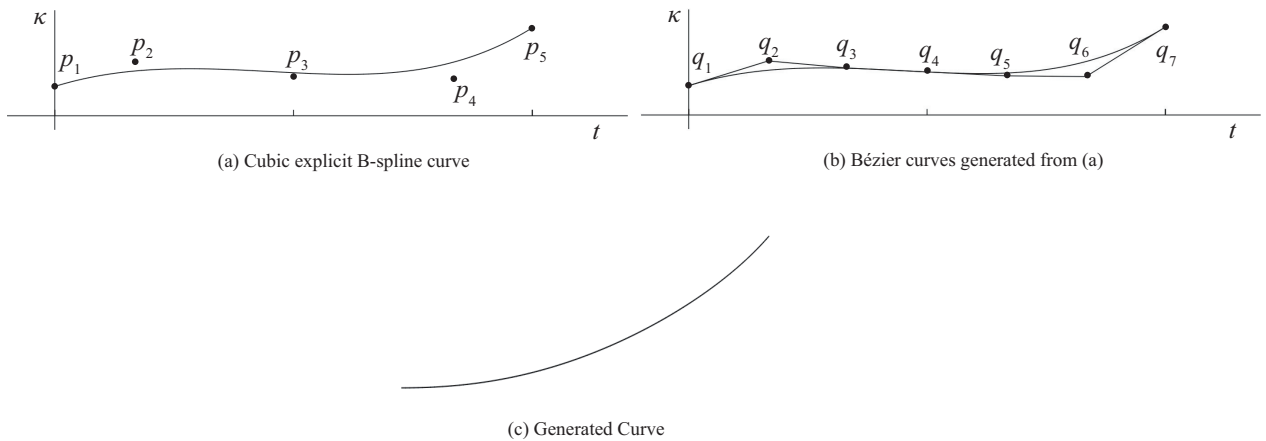


Figure 1: Curvature in terms of an explicit cubic B-spline curve (a), (b) and the generated curve (c)

explicit Bézier curve with the range $t \in [0, 1]$ is just the average of the control points [16], integration of the explicit B-spline curve of Fig. 1(a) is

$$\int_0^1 \sum_{i=1}^5 B_i^3(t) p_i dt = \frac{1}{2} \frac{q_1 + q_2 + q_3 + q_4}{4} + \frac{1}{2} \frac{q_4 + q_5 + q_6 + q_7}{4}. \quad (5)$$

Note that $\frac{1}{2}$ in the right side of Eq. (5) is the parameter range of the first or second explicit Bézier curve segment.

Let s_t be the arc length of a curve segment and $\kappa_i (i = 1, 2, \dots, m)$ be *control curvatures*. Without loss of generality, we assume $t_n = 0$ and $t_m = 1$ to make the parameter range of an explicit B-spline curve $[0, 1]$. The curvature function $\kappa(s)$ in terms of arc length s in the explicit polynomial B-spline form is

$$\kappa(s) = \sum_{i=1}^m B_i^n(s/s_t) \kappa_i \quad (s \in [0, s_t]). \quad (6)$$

Plugging Eq. (6) into Eq. (1) and then Eq. (1) into Eq. (2), curve points in the standard form can be computed. Note that to compute Eq. (1), numerical integration is not necessary. This integration can be computed by converting an explicit B-spline curve to explicit Bézier curves using knot insertion and using the characteristics of explicit Bézier curves [16, 25].

To compute the curve point at arc length $s_u (0 \leq s_u \leq s_t)$, we need to compute the tangential angle $\theta(s_u)$. We are given an explicit B-spline curvature function (Eq. (6)) with arc length s_t . We insert knots at $t_u = \frac{s_u}{s_t}$ at most n times to divide the explicit B-spline curve into two curves with parameter ranges $[0, t_u]$ and $[t_u, 1]$. By appropriately inserting knots, we convert the explicit B-spline curve with the range $[0, t_u]$ to explicit Bézier curves with g_z segments and m_z control curvatures $\hat{\kappa}_i (i = 0, \dots, m_z)$. Let $u_i (i = 1, \dots, g_z)$ be the parameter range of each explicit Bézier curve segment. Now Eq. (1) can be simply computed by

$$\begin{aligned} \theta(s_u) &= \int_0^{s_u} \kappa(u) du \\ &= s_u \sum_{i=1}^{g_z} u_i \frac{\sum_{j=1}^n \hat{\kappa}_{n(i-1)+j}}{n+1}. \end{aligned} \quad (7)$$

Concerning Eq. (2), numerical integration is inevitable. Fig. 1(c) shows the generated curve in the standard form using the curvature function of (a) when the arc length s_t is given.

In an explicit B-spline curvature function, detecting an inflection point and curvature extrema is straightforward. To detect an inflection point, we check if $\kappa(s)$ becomes 0 using Bézier clipping [17]. Similarly, to detect curvature extrema, we need to check if the derivative of $\kappa(s)$ with respect to s becomes 0 using Bézier clipping. For a low degree curvature function, such points can be directly computed.

4 G^1 AND G^2 HERMITE INTERPOLATION METHOD

In G^1 Hermite interpolation, a starting point \mathbf{P}_s and an endpoint \mathbf{P}_e with their unit tangent vectors \mathbf{t}_s and \mathbf{t}_e are given. In G^2 Hermite interpolation, control curvatures κ_s and κ_e at \mathbf{P}_s and \mathbf{P}_e are also given. G^1 and G^2 Hermite interpolation methods works in a similar way except that the first and the last control curvatures are set to κ_s and κ_e , respectively, in case of G^2 Hermite interpolation.

In G^1 or G^2 Hermite conditions, we use the standard form where \mathbf{P}_s is placed at the origin and $\mathbf{t}_s = [1 \ 0]^T$. Let θ_d be the angle formed by \mathbf{t}_s and \mathbf{t}_e . Thus $\mathbf{t}_e = [\cos(\theta_d) \ \sin(\theta_d)]^T$. Given the degree n , the number control curvatures m and $\kappa_i (i = 1, \dots, m)$, the arc length s_t of the curve segment is given by

$$s_t = \frac{\theta_d}{\int_0^1 (\sum_{i=1}^m B_i^n(t)\kappa_i) dt} \tag{8}$$

In G^1 or G^2 Hermite interpolation in the standard form, the generated curve must satisfy the following conditions:

$$\begin{aligned} (a) \quad & \mathbf{P}(0) = \mathbf{P}_s, \quad (b) \quad \mathbf{P}(s_t) = \mathbf{P}_e, \\ (c) \quad & [\cos(\theta(0)) \ \sin(\theta(0))]^T = \mathbf{t}_s, \quad (d) \quad [\cos(\theta(s_t)) \ \sin(\theta(s_t))]^T = \mathbf{t}_e. \end{aligned} \tag{9}$$

Conditions (a), (c) are always satisfied by using Eq. (1) and (2), respectively. Condition (d) is always satisfied by using the arc length s_t computed by Eq. (8). The remaining condition (b) is satisfied by an optimization that minimizes $|\mathbf{P}(s_t) - \mathbf{P}_e|$.

Fig. 2 shows typical cases how the control curvatures are computed. In G^1 Hermite interpolation, we typically use κ_1 and κ_m as optimization parameters and $\kappa_2, \dots, \kappa_{m-1}$ are either user-specified or computed by some kind of interpolation, such as linear interpolation. In G^2 Hermite interpolation, $\kappa_1 (= \kappa_s)$ and $\kappa_m (= \kappa_e)$ are given in addition to G^1 Hermite conditions. Therefore, we typically use κ_2 and κ_{m-1} as optimization parameters and compute $\kappa_3, \dots, \kappa_{m-2}$ similarly by some kind of interpolation or specify them.

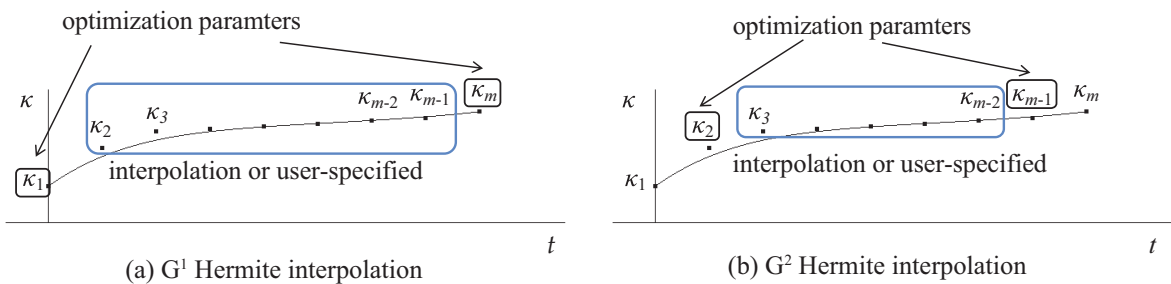


Figure 2: Typical G^1 and G^2 Hermite interpolations

To modify the curvature variation under given Hermite conditions, we use the following interpolation by a parameter $p \in [-1, 1]$ to compute intermediate control curvatures. If $p = 0$ the interpolation becomes the linear

interpolation. See the curvature plots of Fig. 3(a)-(e) to see how the curvature plot changes depending on the value of p . Suppose we are given κ_{i_1} and κ_{i_m} ($1 \leq i_1 < i_m \leq m$). We would like to compute $\kappa_{i_1+1}, \dots, \kappa_{i_m-1}$ using κ_{i_1} , κ_{i_m} and p . Let x_i be the Greville abscissa for κ_i . Then each κ_i ($i = i_1+1, \dots, i_m-1$) is computed by

$$temp = \frac{t_{i_m} - x_i}{t_{i_m} - t_{i_1}} \kappa_{i_1} + \frac{x_i - t_{i_1}}{t_{i_m} - t_{i_1}} \kappa_{i_m} \quad (10)$$

$$\kappa_i = \begin{cases} (1-p) temp + p \kappa_{i_1} & \text{if } p \geq 0 \\ (1+p) temp - p \kappa_{i_m} & \text{if } p < 0 \end{cases} \quad (11)$$

The user can modify the curvature variation of the curve by modifying p .

For typical G^1 Hermite interpolation, we set $i_1 = 1$ and $i_m = m$. The monotonicity of curvature is always guaranteed if $p \in [-1, 1]$, though the curve may include an inflection point. For typical G^2 Hermite interpolation, we set $i_1 = 1$ and $i_m = m - 1$.

5 RESULTS

5.1 Implementation and Computation Time

We have implemented our method in C++. Once G^1 or G^2 Hermite conditions are given, we transform the two endpoints and their tangents such that they satisfy the standard form. More specifically, we perform a translation such that first endpoint is placed at the origin, perform a rotation such that the first tangent become $[1 \ 0]^T$. Then the second endpoint is scaled such that it is on the unit circle. Note that in case of G^2 Hermite interpolation, two given control curvatures are inversely scaled. Under the standard form, appropriate control curvatures are computed and then curve points are computed. The computed curve points are inversely transformed so that they fits the original G^1 or G^2 Hermite conditions. For computing numerical integration and performing an optimization, we use GNU Scientific Library (GSL). We use the downhill simplex method for optimization.

Since we use the same framework for G^1 and G^2 Hermite interpolation, we show the computation time of G^1 Hermite interpolation. Table 1 shows the average computation time of a curve segment and its standard deviation for various degrees and number of control points on Ryzen 9 3950X with 64GB RAM. To measure the computation time, randomly chosen 500 G^1 Hermite conditions are used and the number of curve points is set to 500. Table 1 shows that the proposed curve can be controlled fully interactively.

Table 1: Computation Time

degree	# of control curv.	comp. time (ms)	std. dev. (ms)
1	2	0.27	0.008
2	3	0.32	0.009
3	4	0.37	0.103
3	6	0.49	0.108
3	8	0.60	0.118

5.2 G^1 Hermite Interpolation

Fig. 3 shows examples of generated curves and their curvature plots for typical G^1 Hermite conditions using explicit cubic B-spline curvature functions. κ values of curvature plots are scaled so that the curvature plot fits

in the graph. In Fig.3 (a)-(e), the number of control curvatures is 6 and the knot vector is $[0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1]$ but the value of p is different. In the curvature plot, black circles represent control curvatures $\kappa_1, \dots, \kappa_6$ from left to right. Tick marks in t -axis represent knots t_i . The rightmost tick mark is placed at $t = 1$. Since we use Bézier end conditions, the multiplicity of knots at $t = 0$ and $t = 1$ are set to 3. By modifying the value of p , the curve shape and the curvature plot can be variously controlled for the same G^1 Hermite condition guaranteeing the monotonicity of curvature. Depending on G^1 Hermite conditions, the generated curve may include an inflection point. Fig. 3(f) shows a curve segment with 4 control curvatures the knot vector $[0, 0, 0, 1, 1, 1]$. This explicit B-spline curvature function is exactly the same as an explicit Bézier function. In this example, κ_1 and κ_4 are used as optimization parameters for satisfying G^1 Hermite conditions and κ_2 and κ_3 are set to be equal to κ_1 and κ_4 , respectively. In this way, we can generate a curve segment whose $\frac{d\kappa}{ds} = 0$ at both endpoints.

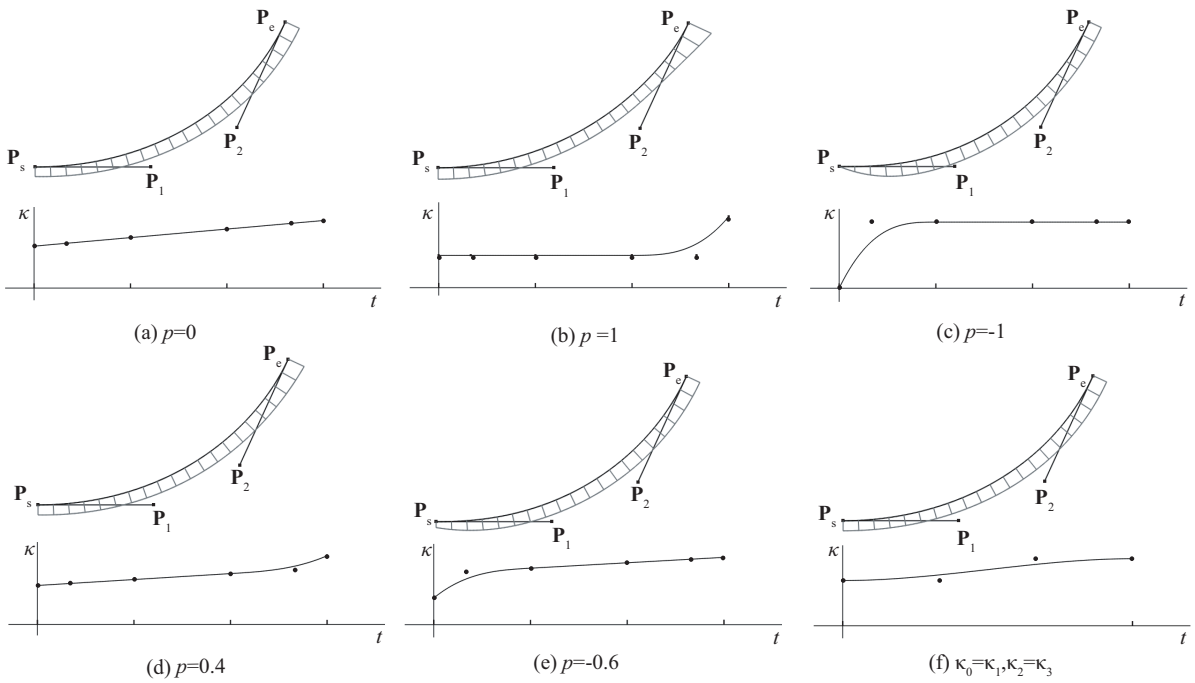


Figure 3: G^1 Hermite interpolation using explicit cubic B-spline curves

5.3 G^2 Hermite Interpolation

In G^2 Hermite Interpolation, we are given κ_s and κ_e in addition to G^1 Hermite conditions. Let θ_e be the angle between $\mathbf{P}_e - \mathbf{P}_s$ and \mathbf{t}_s . See Fig. 4.

Without loss of generality, we consider the case of monotonically increasing curvature without an inflection point. We also assume the curvature is greater than or equal to 0. Thus $0 \leq \kappa_s < \kappa_e$ and $\theta_d > 2\theta_e$. If $\theta_d < 2\theta_e$, curves with monotonically decreasing curvature are generated. For $0 < \theta_e < (\theta_d - \theta_e) < \frac{\pi}{2}$ and $|\mathbf{P}_e - \mathbf{P}_s| = 1$, if the following inequalities[5, 2] are satisfied, there exists a curve segment with monotonically varying curvature satisfying given G^2 Hermite conditions.

$$\kappa_e > \frac{2(1 - \cos \theta_d) - 2 \sin(\theta_d - \theta_e)}{2 \sin \theta_e - \kappa_s}, \quad \kappa_s < 2 \sin \theta_e \tag{12}$$

Note that if $|\mathbf{P}_e - \mathbf{P}_s| \neq 1$, we can replace κ_s and κ_e by $\frac{\kappa_s}{|\mathbf{P}_e - \mathbf{P}_s|}$ and $\frac{\kappa_e}{|\mathbf{P}_e - \mathbf{P}_s|}$, respectively. If θ_e and θ_d are given, the left inequality in Eq. (12) describes a region in (κ_s, κ_e) space above a hyperbola. Fig. 5 shows a hyperbola for $\theta_e = \frac{7}{36}\pi$ and $\theta_d = \frac{4}{9}\pi$. If (κ_s, κ_e) is inside the region and $\kappa_s \geq 0$, there exists some curve segment with monotonically varying curvature satisfying given conditions.

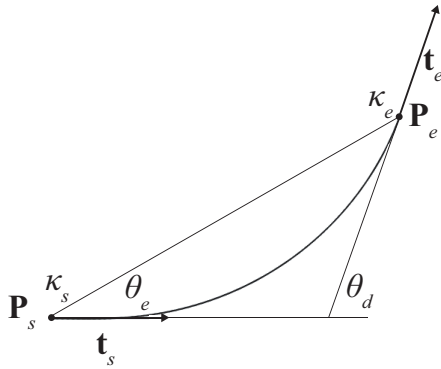


Figure 4: G^2 Hermite interpolation

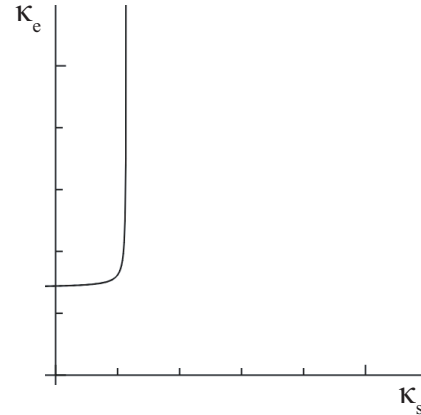


Figure 5: A hyperbola for $\theta_e = \frac{7}{36}$ and $\theta_d = \frac{4}{9}$

For $\theta_e = \frac{1}{9}\pi$, $\theta_d = \frac{1}{3}\pi$ and $|\mathbf{P}_e - \mathbf{P}_s| = 1$, Fig. 6 shows the (κ_s, κ_e) viable regions where curvature is monotonically varying. For (κ_s, κ_e) points satisfying Eq. (12), if a curve with monotonically varying curvature is generated, the point in the (κ_s, κ_e) graph is plotted in black. Otherwise, the point is remain in white. In Fig. 6(a), an explicit cubic B-spline curvature function with 4 control curvatures and knot vector $[0, 0, 0, 1, 1, 1]$ is used. In Fig. 6(b),(c),(d), an explicit cubic B-spline curvature function with 6 control curvatures and knot vector $[0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1]$ is used. The values of p for the interpolation using Eqs. (10), (11) are: (b) $p = 0$, (c) $p = 0.5$, and (d) $p = 1$. In Fig. 6(e), an explicit cubic B-spline curvature function with 6 control curvatures with knot vector $[0, 0, 0, 0.0333, 0.966, 1, 1, 1]$ and $p = 1$ is used. If (κ_s, κ_e) is in the black region, a curve segment with monotonically varying curvature exists satisfying the given G^2 Hermite conditions. Fig. 7 (a) and (b) show examples of curves satisfying the given G^2 Hermite conditions whose (κ_s, κ_e) is in the black region of Fig. 6 (a) and (e), respectively.

Fig. 8 shows examples of G^2 Hermite interpolation for $\theta_e = \frac{1}{9}\pi$, $\theta_d = \frac{1}{3}\pi$ and $|\mathbf{P}_e - \mathbf{P}_s| = 1$. Fig. 8(a) shows the region of (κ_s, κ_e) where curves with monotonically varying curvature can be generated. In Fig. 8 (b),(c),(d),(e), (κ_s, κ_e) are the points denoted as **a**, **b**, **c** and **d** in Fig. 8(a), respectively. The degree and the number of control curvatures are 3 and 8, respectively. The control curvatures and the knots are set manually to make the curvature monotonically varying. Based on these results and other experiments we have performed, proposed curves can match a wide variety of G^2 Hermite conditions by appropriately specifying the number of control points, control curvatures and a knot vector. Especially for the pairs of (κ_s, κ_e) denoted as **a**, **b** and **c**, G^2 Hermite interpolation with monotonically varying curvature is impossible by the previous method[25].

5.4 Controlling the Curvature Variation on the Curvature Comb

Using an explicit B-spline curvature function provides a method to control the curvature variation by directly modifying control curvatures on the curvature comb. This section shows examples of controlling the curvature variation using explicit cubic B-spline curvature functions with 6 control curvatures.

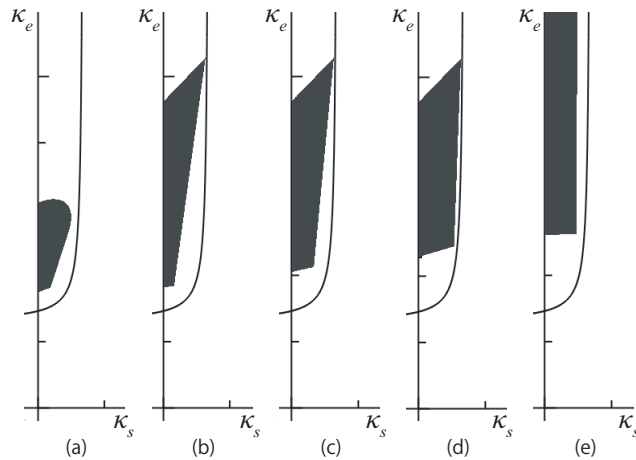


Figure 6: (κ_s, κ_e) viable regions

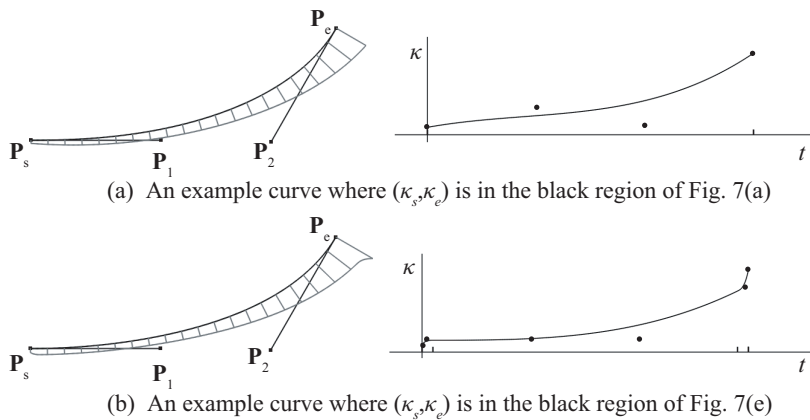


Figure 7: G^2 Hermite interpolation examples

Fig. 9 (a) shows a curve segment and its curvature plot satisfying given G^1 Hermite conditions. On the curvature comb, appropriately scaled control curvatures are plotted as black dots in the normal direction of the curve. In Fig. 9 (b), control curvature κ_4 indicated by an arrow is moved. To control the curvature variation, the control curvature can be interactively moved in the normal direction of the curve. To modify κ_4 , the scaled (signed) difference between the original position of κ_4 and the moved position in the normal direction of the curve is added to the original κ_4 . After κ_4 is modified, an optimization is performed to satisfy the given G^1 Hermite conditions with κ_4 fixed. In this example, κ_1 and κ_6 are used for optimization parameters. In Fig. 9 (c), control curvature κ_5 indicated by an arrow is further moved. Note that large movement of a control curvature may generate an undesirable curve shape including inflection points and curvature extrema, or may not satisfy given G^1 Hermite conditions. Controlling a control curvature can also be done in the curvature plot.

Fig. 10 (a) shows G^1 curve segments and their curvature plots. In Fig. 10 (b), control curvatures indicated by arrows are interactively moved to make the curve segments G^2 continuous. Modifying control curvatures can also be used to make curve segments G^2 continuous.

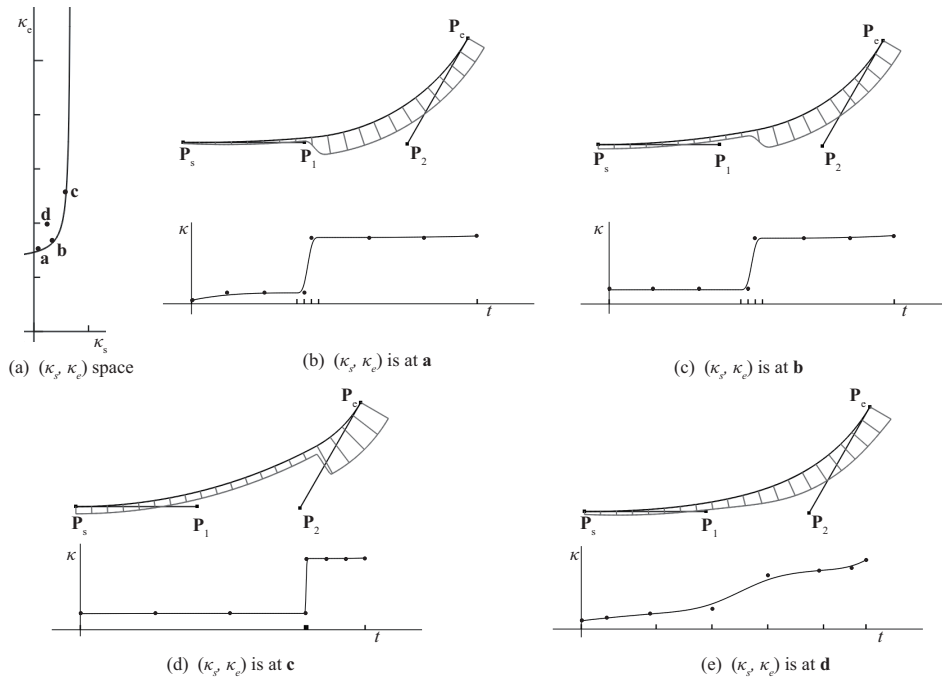


Figure 8: Examples of G^2 Hermite interpolation

6 CONCLUSIONS

In this paper, we proposed intrinsically defined curves based on explicit B-spline curvature functions. In the proposed method, two of the control curvatures are used as an optimization for fitting G^1 or G^2 Hermite interpolation conditions and other control curvatures (except for two control curvatures specified in case of G^2 Hermite interpolation) are either interpolated or user-specified. Given G^1 or G^2 Hermite interpolation conditions, the proposed curve has a wider variety of curvature variation than our previous work[25] due to the use of an explicit B-spline curvature function. We also proposed a method for interactively controlling the curvature variation by moving the control curvature shown on the curvature comb or curvature plot. Future work includes a method for automatically covering all the viable region of G^2 Hermite interpolation and the extension to space curves

Norimasa Yoshida, <http://orcid.org/0000-0001-8889-0949>

Takafumi Saito, <http://orcid.org/0000-0001-5831-596X>

REFERENCES

- [1] Ali, J.M.; Tookey, R.M.; Ball, J.V.; Ball, A.A.: The generalised Cornu spiral and its application to span generation. *Journal of Computational and Applied Mathematics*, 7, 1999. [http://doi.org/10.1016/S0377-0427\(98\)00207-6](http://doi.org/10.1016/S0377-0427(98)00207-6).
- [2] Dietz, D.A.; Piper, B.: Interpolation with cubic spirals. *Computer Aided Geometric Design*, 21(2), 2004. <http://doi.org/10.1016/j.cagd.2003.09.002>.
- [3] Dietz, D.A.; Piper, B.; Sebe, E.: Rational cubic spirals. *Computer Aided Design*, 40(1), 3–12, 2008. <http://doi.org/10.1016/j.cad.2007.05.001>.

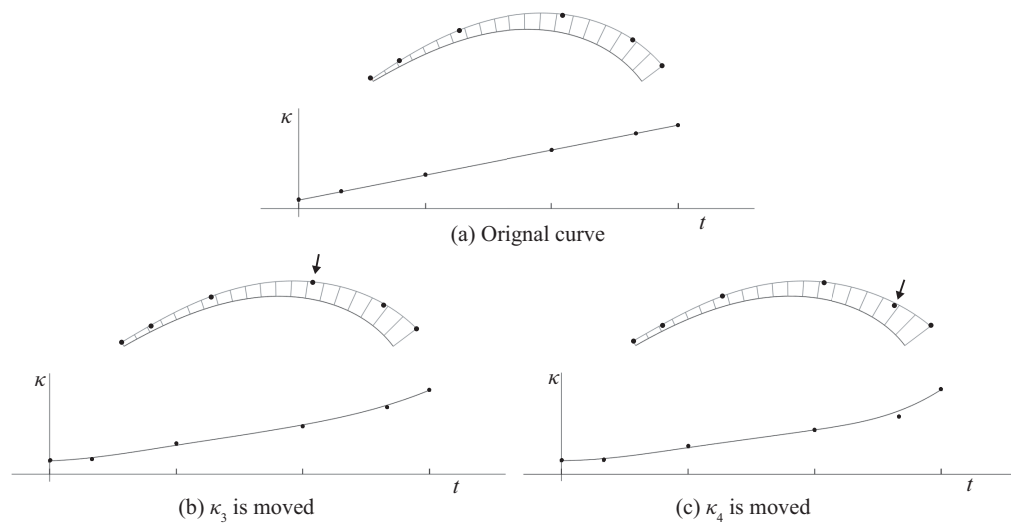


Figure 9: Direct control of control curvatures on the curvature comb

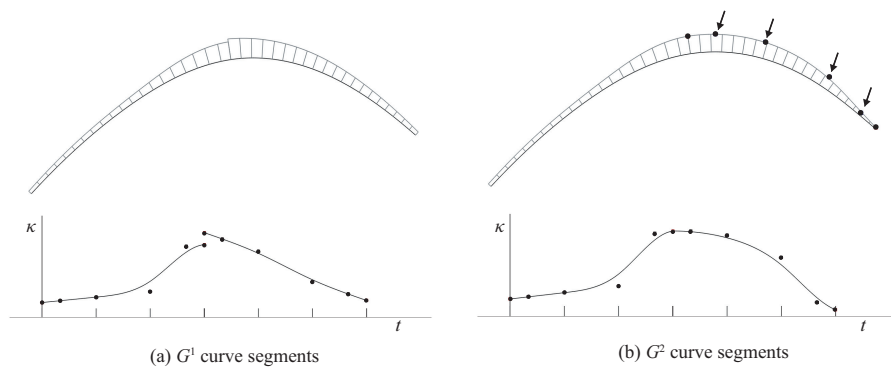


Figure 10: Making G^1 curve segments to G^2

- [4] Farin, G.: Class A Bézier curves. *Computer Aided Geometric Design*, 23(7), 573–581, 2006. <http://doi.org/10.1016/j.cagd.2006.03.004>.
- [5] Guggenheimer, H.W.: *Differential Geometry*. Dover, 1997.
- [6] Habib, Z.; Sakai, M.: Admissible regions for rational cubic spirals matching G^2 Hermite data. *CAD Computer Aided Design*, 42(12), 1117–1124, 2010. <http://doi.org/10.1016/j.cad.2010.07.006>.
- [7] Harada, T.; Yoshimoto, F.; Moriyama, M.: An aesthetic curve in the field of industrial design. In *Proceedings of IEEE Symposium on Visual Languages*, 38–47. IEEE Computer Society Press, New York, 1999. <http://doi.org/10.1109/VL.1999.795873>.
- [8] Li, W.; Xu, S.; Zheng, J.; Zhao, G.: Target curvature driven fairing algorithm for planar cubic b-spline curve. *Computer Aided Geometric Design*, 21, 499–513, 2004. <http://doi.org/10.1016/j.cagd.2004.03.004>.
- [9] Meek, D.S.: Planar spirals that match G^2 Hermite data. *Computer Aided Geometric Design*, 15(2), 103–126, 1998. [http://doi.org/10.1016/S0167-8396\(97\)00020-4](http://doi.org/10.1016/S0167-8396(97)00020-4).

- [10] Miura, K.T.: A general equation of aesthetic curves and its self-affinity. *Computer-Aided Design and Applications*, 3(1–4), 457–464, 2006. <http://doi.org/10.1080/16864360.2006.10738484>.
- [11] Miura, K.T.; Shibuya, D.; Gobithaasan, R.U.; Usuki, S.: Designing Log-aesthetic Splines with G^2 Continuity. *Computer-Aided Design and Applications*, 10(6), 1021–1032, 2013. <http://doi.org/10.3722/cadaps.2013.1021-1032>.
- [12] Nutbourne, A.W.; McLellan, P.M.; Kensit, R.M.L.: Curvature profiles for plane curves. *Computer-Aided Design*, 4(4), 176–184, 1972. [http://doi.org/10.1016/0010-4485\(72\)90072-3](http://doi.org/10.1016/0010-4485(72)90072-3).
- [13] Pal, T.K.: Intrinsic spline curve with local control. *Computer-Aided Design*, 10(1), 19–29, 1978. [http://doi.org/10.1016/0010-4485\(78\)90005-2](http://doi.org/10.1016/0010-4485(78)90005-2).
- [14] Ramshaw, L.: Blossoms are polar forms. *Computer Aided Geometric Design*, 6(4), 1989. [http://doi.org/10.1016/0167-8396\(89\)90032-0](http://doi.org/10.1016/0167-8396(89)90032-0).
- [15] Sapidis, N.S.; Frey, W.H.: Controlling the curvature of a quadratic Bézier curve. *Computer Aided Geometric Design*, 9(2), 85–91, 1992. [http://doi.org/10.1016/0167-8396\(92\)90008-D](http://doi.org/10.1016/0167-8396(92)90008-D).
- [16] Sederberg, T.W.: Computer aided geometric design, 2017. URL: <http://hdl.lib.byu.edu/1877/2822>.
- [17] Sederberg, T.W.; Nishita, T.: Curve intersection using bézier clipping. *Computer-Aided Design*, 22(9), 1990. [http://doi.org/10.1016/0010-4485\(90\)90039-F](http://doi.org/10.1016/0010-4485(90)90039-F).
- [18] Tsuchie, S.; Okamoto, K.: High-quality quadratic curve fitting for scanned data of styling design. *CAD Computer Aided Design*, 71, 39–50, 2015. ISSN 00104485. <http://doi.org/10.1016/j.cad.2015.09.004>.
- [19] Walton, D.J.; Meek, D.S.: G^1 interpolation with a single Cornu spiral segment. *Journal of Computational and Applied Mathematics*, 223(1), 86–96, 2009. ISSN 03770427. <http://doi.org/10.1016/j.cam.2007.12.022>.
- [20] Wang, Y.; Zhao, B.; Zhang, L.; Xu, J.; Wang, K.; Wang, S.: Designing fair curves using monotone curvature pieces. *Computer Aided Geometric Design*, 21(5), 515–527, 2004. ISSN 01678396. <http://doi.org/10.1016/j.cagd.2004.04.001>.
- [21] Watanabe, Y.; Saito, T.; Kuroda, M.: A method for generation of curves from specified curvature profile (*in japanese*). In *Graphics and CAD Workshop of Information Processing Society of Japan*, 7–12, 1997.
- [22] Yan, Z.; Schiller, S.; Wilensky, G.; Carr, N.; Schaefer, S.: κ -Curves : Interpolation at Local Maximum Curvature. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 36(4), 129:1—129:7, 2017. <http://doi.org/10.1145/3072959.3073692>.
- [23] Yoshida, N.; Hiraiwa, T.; Saito, T.: Interactive Control of Planar Class A Bézier Curves using Logarithmic Curvature Graphs. *Computer-Aided Design and Applications*, 5(1-4), 121–130, 2008. ISSN 1686-4360. <http://doi.org/10.3722/cadaps.2008.121-130>.
- [24] Yoshida, N.; Saito, T.: Interactive aesthetic curve segments. *The Visual Computer*, 22(9-11), 2006. <http://doi.org/10.1007/s00371-006-0076-5>.
- [25] Yoshida, N.; Saito, T.: Planar curves based on explicit bézier curvature functions. *Computer Aided Design and Applications*, 17(1), 2020. <http://doi.org/10.14733/cadaps.2020.77-87>.